# Visual Storytelling with Semantic Reasoning

**Chandan Akiti**
The Pennsylvania State University
University Park, Pennsylvania
cra5302@psu.edu

**Ting-Yao Hsu**
The Pennsylvania State University
University Park, Pennsylvania
txh357@psu.edu

**Pratheebha Karuppusamy**
The Pennsylvania State University
University Park, Pennsylvania
pxk227@psu.edu

## ABSTRACT

Vision-Language models, an integration of computer vision and natural language processing are widely used across various disciplines, one such application is visual storytelling. Since modern visual storytelling models are still hard to generate a sound story, we propose a hybrid model inspired by previous works [9, 6]. This model combines visual semantic representations and global-local attention mechanisms to generated narrative stories. Our model separates into two parts. First we make a connection between image regions and use Graph Convolutional Networks to generated semantic relationships. Then use simple Gated Recurrent Network to select the essential information for the whole image. Second, we incorporated two levels of attention, one is sequential images encoding, another is regional image features to generated sentences. We aim to improve one of state-of-the-art story generation model by capturing semantic concepts of the salient parts in the scene. We evaluate our hybrid model on the visual storytelling dataset (VIST) and show the story quality and coherence can be improved comparing to state-of-the-art techniques.

## INTRODUCTION

Vision-language models combine information obtained from both vision and natural language to have an in-depth thorough analysis of visual scenes. The recent evolution of computer vision and natural language processing algorithms have put vision-language models to a remarkable use in a wide range of applications, such as image captioning, visual question answering, image and video retrieval, visual storytelling and so on.

Visual storytelling [15] is the specific task in the field, which generates a cohesive narrative and describes the various events happening across the sequence of images. This task has the capability to change from basic understandings of visual scenes towards more human-like understanding of grounded event structure and subjective expression.

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

In this project, we introduce a hybrid deep learning architecture to generate visual stories that combines global and local attention of sequential images. We (2) further apply semantic reasoning concepts to enhance regional relationships and rule out unimportant information of the sequential images. Our model shows that build up a regional semantic relationship can further improve the story coherence and quality.

## RELATED WORK

Image captioning is a well studied topic in the vision-to-language problem [4, 13, 18, 19, 20, 17]. In the previous work, Vinyals *et al.* [16] and Xu *et al.* [19] used CNN-based images encoder and RNN-based for decoding. Attention-based neural networks learns to focus on different parts of the image are also well-used to generate descriptive captions [5, 1]. Several works use reinforcement learning-based methods on image captioning system. Rennie *et al.* [14] used the test-time inference algorithm to modify the reward rather than in training time. Zhang *et al.* proposed an actor-critic reinforcement learning method to optimize non-differentiable problems of the existing evaluation metrics. And Ren *et al.* [13] proposed a new architecture that two networks can jointly compute the next word at each time step.

There are several existing visual storytelling models. One of the state-of-the-art achieved the highest human evaluation score in the first VIST Challenge [11] was GLAC (Global-Local Attention Cascading Networks) [6]. This model was designed to generate *sequential captions for the given input image sequence*. The novelty is the attention cascading of local image feature with global context of image in the sequence. Thus, producing the caption for that image in coherent to the story for the given sequence image.

Another interesting work in single image captioning is VSRN (Visual Semantic Reasoning for Image-Text Matching) [9]. This model introduced semantic reasoning concept for *single* image captioning. The models takes the most salient regions in an image, and region relationship is learnt through Graph Convolution Networks. These enhanced region embeddings are used to create a representation of image. This novelty is interesting because the saliency consideration is in itself a way of attention.

## DATASET

The sequential vision-to-language dataset used in this project is the Visual storytelling(VIST) [15] dataset. This dataset consists of 210,819 unique photos from Flickr, which are arranged in different ways to form sequences with 5 images

per sequence. These sequences are aligned to both descriptive and narrative language with 3 tiers of language description for each sequence viz.,

1. Description of images-in-isolation(DII)
2. Description of images-in-sequence(DIS)
3. **Stories for images-in-sequence(SIS)**

For the visual storytelling task, we use the stories for images-in-sequence descriptions. The VIST dataset is modeled in such a way that it has the potential to design algorithms that can make progress from basic understanding of visual scenes towards more human like understanding of the scene. Figure 1 shows an example sequence from the dataset.

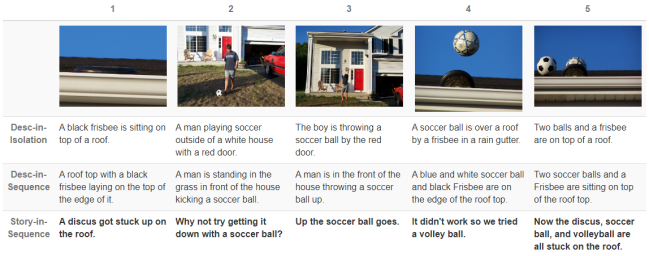


**Figure 1. One photo sequence of the VIST Dataset**

## PROPOSED METHOD

Our hypothesis is that, we can increase the relevance of captions generated by learning semantic relations between the parts of image. We hope to accomplish this by using Graph Convolution Networks.

Motivated by our hypothesis, we propose the following model. This model would be an enhancement on encoder of GLAC [6] model.

The goal is to infer the similarity between the five generated caption sentences and a whole story of five images in sequence by mapping image regions and text descriptions into a common embedding space. We divide the model into four parts.

**Pre-processing** Here we extract salient regions from the images using bottom-up-attention model [1].

**Encoder** Build connections between these image regions and do reasoning using Graph Convolution Networks (GCN) [7] to generate features with semantic relationship information. We do global semantic reasoning on these relationship-enhanced features to select discriminative information and filter out unimportant ones to generate the final representation for the whole image.

**Global Attention** For the task of story generation, the caption generated for each image should not only be grounded by the image itself, but also the global context of the story that is the sequence of five images. We generate global contextual feature from the local semantic reasoning features of images using a Bi-LSTM [3]. Here we could use LSTM too, but Bi-LSTM is better for aggregating global context.

**Decoder** Concatenation of the global feature with local feature of each image would be the input to the decoder LSTM. During the training process, we generate captions with same length as target captions. During the inference process, we generate until we get a token *'<end>'* or up to 50 words.

### Pre-processing - Bottom-Up Attention

We use bottom-up attention model [1] for the pre-processing step. This pre-trained model generates output features of salient image regions. These bottom-up attention features can typically be used as a drop-in replacement for CNN features in attention-based image captioning and visual question answering (VQA) models. This approach was used to achieve state-of-the-art image captioning performance on MSCOCO (CIDEr 117.9, BLEU_4 36.9) dataset [10] and to win the 2017 VQA Challenge (70.3% overall accuracy).

This model is implemented by Faster R-CNN which is an object detection model designed to identify instances of objects belonging to certain classes and localize them with bounding boxes. Faster R-CNN detects objects in two stages. We only use the first stage, described as a Region Proposal Network (RPN), to extract salient region proposals.

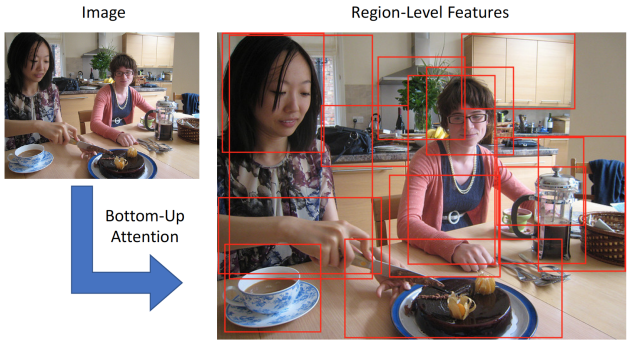


**Figure 2. Use Bottom-Up Attention to extract the local features from top $k$ boxes in each image**

In a typical Faster R-CNN architecture, a small network is slid over features at an intermediate level of a CNN. At each spatial location the network predicts a class-agnostic objectness score and a bounding box refinement for anchor boxes of multiple scales and aspect ratios. Using greedy non-maximum suppression with an intersection-over-union (IoU) threshold (0.3), the top box proposals are selected.

For the purpose of our project, we select top $k$ bounding boxes from the boxed produced by Region Proposal Network based on the sorted confidence score. This is same as two previous papers [8] [12] where they already proved the setting will get the best results. We set $k = 36$ as it is a good number to cover the major part of image. A low $k$ will miss some salient parts that might add context to the story. A higher $k$ will include bounding boxes with low confidence score. Each bounding box is represented by a 2048 feature vector. Finally, $k \times 2048$ feature for the image is extracted and saved for the training purpose. (See Figure 2).

### Encoder and Global Attention

Inspired by the VSRN [9] paper, we also use Graph Convolution Networks (GCN) to construct region relationship reasoning (See Figure 3.) and pass it through GRUs [2] to perform semantic reasoning. First, we build up a region relationship reasoning model to enhance the semantic correlation between image regions. We consider each image region as a vertex $v_i$ of the graph. Pairwise affinity between the image regions in an embedding space is calculated to show their relationship

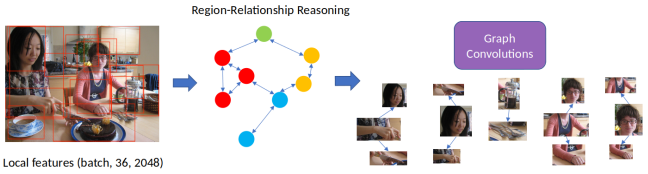


**Figure 3. For each image, Construct a graph with $k$ nodes. Each node $v_i$ denotes a bounding box feature of an image. An edge is defined as the affinity as in eq 1. Apply GCN to perform region relationship reasoning as in eq 2**

Graph $G_r = (V, E)$ is a fully-connected relationship graph, $V$ is the set of image regions and $E$ is the set decided by the affinity matrix $R$.

$$R(v_i, v_j) = \varphi(v_i)^T \phi(v_j) \tag{1}$$

where $\varphi(v_i) = W_\varphi v_i$ and $\phi(v_j) = W_\phi v_j$ are two embeddings.

We then apply GCN to perform semantic reasoning on this fully connected network and add residual connections to the it.

$$V^* = W_r(RVW_g) + V \tag{2}$$

where $W_r$ is residual weight matrix, $W_g$ is weight matrix of GCN layer. $R$ is the affinity matrix with shape of $k * k$.

The weight matrices $W_r$, $W_g$, $W_\varphi$ and $W_\phi$ are learned though back propagation. Thus each GCN layer is a fully connected graph with edges as an affinity that can be calculated by $W_\varphi$ and $W_\phi$. The output of $V^* = \{v_1^*, ..., v_k^*\}$ is the relationship enhanced representation for image regions.

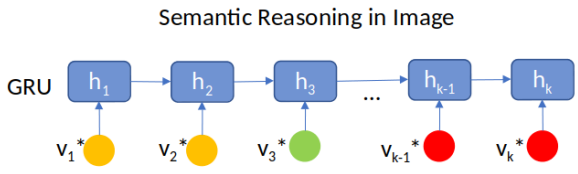


**Figure 4. For each image, perform semantic reasoning with GRU between the bounding boxes. we take the last hidden state of GRU $h_k$ as final representation of image**

Second, we apply the regional features $V^*$ into GRUs one by one to perform global semantic reasoning (See Figure 4). This step will help the network to decide which information to filter out and obtain the final representation of the whole image.

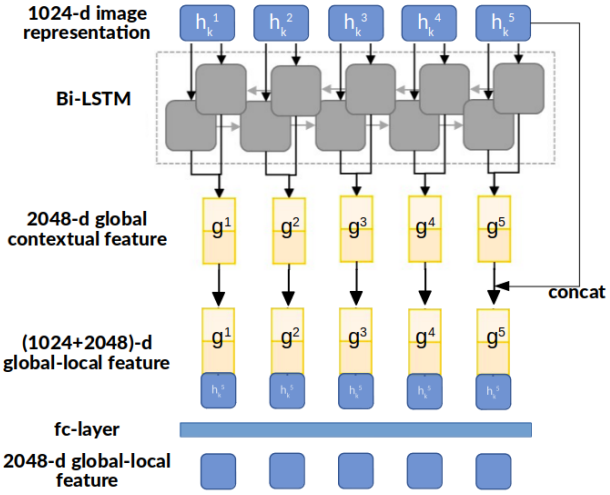


**Figure 5. $\{h_k^i\}$ are the representation of images (local) in sequence. $\{g^i\}$ are the representations of images in global context of sequence. The final 2048-d global-local feature after dropout and batch normalization is sent to the decoder along with the input captions.**

.

At each reasoning step $i$, an update gate $z_i$ analyses the input feature $v_i^*$ and whole scene from last step $h_{i-1}$.

$$z_i = \sigma(W_z v_i^* + U_z h_{i-1} + b_z)$$

where $\sigma$ is a sigmoid activation function. $W_z$, $U_z$ and $b_z$ are weights and bias.

Reset gate $r_i$ analyzes the what content to forget based on the reasoning between $v_i^*$ and $m_{i-1}$.

$$r_i = tanh(W_r v_i^* + U_r h_{i-1} + b_r)$$

where $\sigma$ is a sigmoid activation function. $W_r$, $U_r$ and $b_r$ are weights and bias.

Output from GRU cell based on the scene till step $i$ is,

$$h_i = (1 - z_i) \circ h_{i-1} + z_i \circ tanh(W_h v_i^* + U_h(r_i \circ h_{i-1}) + b_h)$$

where $W_h$, $U_h$ and $b_h$ are weights and bias.

Finally, after $k$ steps, each step corresponding to the salient region, we get representation of whole scene in last state, $h_k$. Thus, for five images in the sequence, we obtain $(h_k^1, h_k^2, h_k^3, h_k^4, h_k^5)$ representations. The dimension of $h_k^j$ is 1024 in our case.

We consider five representations as a sequence and get Bi-LSTM outputs for each image. (See Figure 5.) Bi-LSTM is well-known for better aggregated representation of sequential images. Then we feed both output of Bi-LSTM (includes all the information of sequential images, which represent global) and each image features (local) to the decoders. From Bi-LSTM we get five outputs $g^i$ for each image with dimension $2 \times 1024 = 2048$. This represents the context of the story of the five images in the sequence.

Now we concatenate the local features with global feature and produce $([h_k^1, g^1], [h_k^2, g^2], [h_k^3, g^3], [h_k^4, g^4], [h_k^5, g^5])$ each with

dimension (1024+2048) as the five features for each image. This is passed to a fully connected later to get 2048 dimension feature and then pass it through dropout layer with dropout probability of 0.5 and apply batch normalization. This makes sense because we are trying to generate a feature based on five images, and we wouldn't want to introduce variance every time a new set of scenes comes in the batches.

### Decoder - LSTM

We pass the above 2048-dimension feature again through a fully connected layer to get a feature of 1024 dimension. Our decoder consists of an embedding layer of dimension 256, an LSTM layer with hidden size of 1024 dimension and a fully connected layer $fc_{vocab}$ mapping to dimension of size of vocabulary. We use decoder during training and inference as follows.

#### *Training*

The tokens $\langle start \rangle$ and $\langle end \rangle$ are added at the front and beginning respectively to the tokenized input captions to train the start and end of sequence. While training, we pass the embedding of tokenized input captions one by one along with the global-local feature extracted from encoder step to get the embedding of next word. We decode the index of word from this embedding with the help of the fully connected layer $fc_{vocab}$ and softmax.

Input size of LSTM is (embedding size + feature size), that is (256 + 1024). Then we pass the previous state of LSTM iteratively as the current state of LSTM. Thus, we generate the next word embedding based on the current input word and the previous state.

$$w_i, h_i = LSTM([input_{i-1}, G], h_{i-1})$$

where $w_i$ is the $i^{th}$ word embedding, $h_i$ is the $i^{th}$ hidden state, $input_{i-1}$ is the $(i-1)^{th}$ input word embedding, and $G$ is the feature vector of image from encoder.

#### *Inference*

For the purpose of inference, we use the first word as the embedding of token $\langle start \rangle$ and iteratively generate next word in the same way as training. We do this until we generate the token $\langle end \rangle$ or up to 50 words.

$$w_i, h_i = LSTM([w_{i-1}, G], h_{i-1})$$

where $w_0 = embed(\langle start \rangle)$, $w_i$ is the $i^{th}$ word embedding, $h_i$ is the $i^{th}$ hidden state, and $G$ is the feature vector of image from encoder.

### Training Objective

The objective of this network is to minimize cross-entropy loss between generated output and target caption of all the images in the sequence.

$$L = \sum_{i=1}^{5} CE(o,t)$$

where $o$ is the output caption and $t$ is the target caption. The words in the captions are tokenized versions with *nltk* library.

### Training

VIST [15] data comes with train-val-test split of 80-10-10%. There are 40,155 sequences for training, 4990 for validation, and 5055 for testing. Each sequence in the story consists of 5 images. Each story is annototated with a story caption.

The training parameters are as follows: Word embedding size for Decoder LSTM is taken as 256. Hidden size for the Encoder RNN, Decoder RNN, and GCN is taken as 1024. The learning rate is set to 0.001 and weight decay is set to $1e-5$. We trained for 30 epochs with batch size of 64.

We trained for only 30 epochs because of time constraints. The dataset is large and we plan to produce complete results in future work.

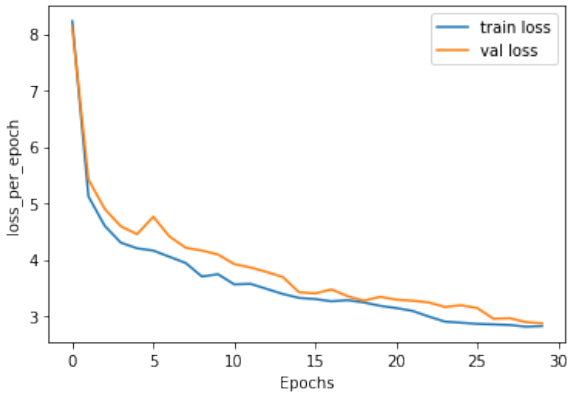


**Figure 6. Training loss plot for 30 epochs. Convergence has not been achieved yet. The model can still performed better if trained longer.**

## EXPERIMENTS

We performed two experiments with different encoders. One model with GCN-GRU-BiLSTM as encoder. Another model with a CNN-BiLSTM as an encoder. Our motivation is to show that GCN-GRU is able to learn better region relationships than using a plain CNN.

The evaluation tool is provided by Storytelling Workshop [11]. The tools provides METEOR, BLUE, and ROUGE scores.

In the experiments with CNN-BiLSTM, we trained the model for 30 epochs only. The training loss converges along with validation loss. The METEOR score for this case is **0.28**.

In the experiment with GCN, we trained for 30 epochs again. But the training loss has not converged yet. The model seems to work pretty well even the training is unfinished. We get a METEOR score of **0.297**. We suspect the model will give better performance if trained for 100 epochs or until convergence. The original GLAC model when trained for 30 epochs gives a score of only 0.263.

We show some of the results of our model in Figure 7, 8, 9 and 10.

## CONCLUSION AND FUTURE WORK

The models developed for the visual storytelling task did the task of generating a story-like caption but did not take into

| Evaluation Metrics | | | |
|---|---|---|---|
| *Model* | *METEOR* | *BLEU1* | *ROUGE* |
| GLAC Net | 0.30 | 0.372 | 0.26 |
| GLAC Net (epoch=30) | 0.263 | 0.339 | 0.216 |
| GLAC Net - CNN (Bounding boxes:36) | 0.28 | 0.365 | 0.28 |
| GLAC Net - GCN (Bounding boxes:36) (epoch=30) | **0.297** | **0.379** | **0.238** |

**Table 1. Automatic evaluation on the VIST dataset. For the first row and third rows we trained for 100 epochs.**


**Figure 7. Our model detected family event and zoo while GLAC model detects it as public event**


**Figure 8. Our model failed here. GLAC was able to detect the carnival. But our model detects it as fireworks event**


**Figure 9. The caption for the first image is about a man. But picture of man doesn't come until 4th image. Our model is able to encode this information successfully.**

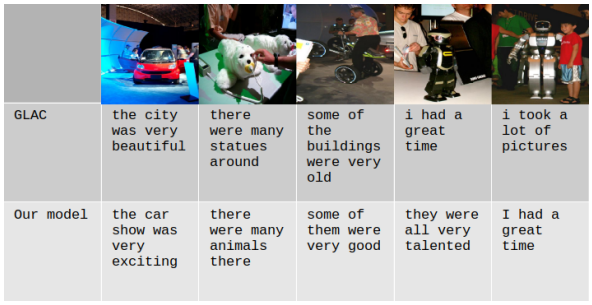


**Figure 10. Another example where our model performs better. Car show is accurately detected by our model. but GLAC predicts it as a beautiful city outdoor event.**

consideration the region relationship and interaction between the images and image regions, thus lacking reasoning in the generated captions. In our model, we devise a GCN enhanced image representation that can capture the key objects in an image as well as the semantic relationships and concepts of the scene.

For this work, we have trained for 30 epochs only on our model because of time constraints created by huge data. We would like to run further experiments and also create an analysis on how Graph Convolution Networks are impacting our model.

## ACKNOWLEDGMENTS

This project was carried out as a part of the special topics course IST597 Fundamentals of Deep Learning offered by The Pennsylvania State University, University Park. The authors would like to thank the instructor Dr. C. Lee Giles and the teaching assistants Ankur Mali and Kaixuan Zhang for their valuable inputs while working on this project.

## REFERENCES

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2017. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. (2017).

[2] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[3] Zixiang Ding, Rui Xia, Jianfei Yu, Xiang Li, and Jian Yang. 2018. Densely connected bidirectional lstm with applications to sentence classification. In *CCF*

*International Conference on Natural Language Processing and Chinese Computing*. Springer, 278–287.

[4] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, and others. 2015. From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1473–1482.

[5] Kun Fu, Junqi Jin, Runpeng Cui, Fei Sha, and Changshui Zhang. 2016. Aligning where to see and what to tell: image captioning with region-based attention and scene-specific contexts. *IEEE transactions on pattern analysis and machine intelligence* 39, 12 (2016), 2321–2334.

[6] Taehyeong Kim, Min-Oh Heo, Seonil Son, Kyoung-Wha Park, and Byoung-Tak Zhang. 2018. GLAC Net: GLocal Attention Cascading Networks for Multi-image Cued Story Generation. *arXiv preprint arXiv:1805.10973* (2018).

[7] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. (2016).

[8] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 201–216.

[9] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. 2019. Visual Semantic Reasoning for Image-Text Matching. (2019).

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

[11] Margaret Mitchell, Ting-Hao Huang, Francis Ferraro, and Ishan Misra. 2018. Proceedings of the First Workshop on Storytelling. In *Proceedings of the First Workshop on Storytelling*.

[12] Will Norcliffe-Brown, Stathis Vafeias, and Sarah Parisot. 2018. Learning conditioned graph structures for interpretable visual question answering. In *Advances in Neural Information Processing Systems*. 8334–8343.

[13] Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. 2017. Deep reinforcement learning-based image captioning with embedding reward. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 290–298.

[14] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7008–7024.

[15] Ting-Hao, Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh, Lucy Vanderwende, Michel Galley, and Margaret Mitchell. 2016. Visual Storytelling. (2016).

[16] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3156–3164.

[17] Cheng Wang, Haojin Yang, Christian Bartz, and Christoph Meinel. 2016. Image captioning with deep bidirectional LSTMs. In *Proceedings of the 24th ACM international conference on Multimedia*. ACM, 988–997.

[18] Yufei Wang, Zhe Lin, Xiaohui Shen, Scott Cohen, and Garrison W Cottrell. 2017. Skeleton key: Image captioning by skeleton-attribute decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7272–7281.

[19] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*. 2048–2057.

[20] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4651–4659.